REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
August 2011	Conference Paper – Post Print	May 2009 – November 2010

4. TITLE AND SUBTITLE
DELAY BASED ROUTING FOR REAL-TIME TRAFFIC IN AD HOC NETWORKS

5a. CONTRACT NUMBER In-House
5b. GRANT NUMBER N/A
5c. PROGRAM ELEMENT NUMBER 62702F

6. AUTHOR(S)
Dipika Darshana, Mainak Chatterjee, and Kevin Kwiat

5d. PROJECT NUMBER 23G4
5e. TASK NUMBER IH
5f. WORK UNIT NUMBER 01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFRL/Information Directorate
Rome Research Site/RIGD
525 Brooks Road
Rome NY 13441

University of Central Florida
Dept of Electrical Engineering & Computer Science
Orlando FL 32816

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/Information Directorate
Rome Research Site
26 Electronic Parkway
Rome NY 13441

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI/RRS

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2011-5

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA #: 88ABW-2010-3301.
Date Cleared: June 16, 2010.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This paper proposes a routing protocol in an ad hoc network that ensures timely delivery of real-time data packets. This is achieved by carefully accessing the resources available to a route before a session is admitted along that route. Each link in the route is checked for sufficient bandwidth not only for the new session to be admitted but also for the sessions that are already using that link. The new session is admitted only if its admission does not violate the delay bounds of any of the on-going sessions. This method of route selection coupled with per-hop link reservations allows the providing of bounds on the delay performance. Extensive simulation experiments are also conducted that demonstrate the performance of the proposed routing protocol in terms of throughput, session blocking probability, average path length and delay.

15. SUBJECT TERMS
Ad Hoc Networks, Routing, Real-time flows, M/M/1 Model

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	7	Kevin Kwiat

19b. TELEPHONE NUMBER (*Include area code*)
N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Delay Based Routing for Real-time Traffic in Ad hoc Networks

Dipika Darshana and Mainak Chatterjee
Department of EECS
University of Central Florida
Orlando, FL 32816
Email: {dipika,mainak}@eecs.ucf.edu

Kevin Kwiat
Air Force Research Laboratory
Information Directorate
Rome, NY 13441
Email: kevin.kwiat@rl.af.mil

***Abstract*—In this paper, we propose a routing protocol in an ad hoc network that ensures timely delivery of real-time data packets. This is achieved by carefully accessing the resources available to a route before a session is admitted along that route. Each link in the route is checked for sufficient bandwidth not only for the new session to be admitted but also for the sessions that are already using that link. The new session is admitted only if its admission does not violate the delay bounds of any of the on-going sessions. This method of route selection coupled with per-hop link reservations allows us to provide bounds on the delay performance. Extensive simulation experiments are also conducted that demonstrate the performance of the proposed routing protocol in terms of throughput, session blocking probability, average path length and delay.**

Keywords: Ad hoc networks, routing, real-time flows, M/M/1 model[1].

I. Introduction

A wireless ad hoc network consists of inexpensive nodes that are randomly deployed over an area of interest forming a self-organizing and self-configuring multi-hop network [4]. These networks do not have any infrastructure support and in most cases are short lived. Military applications and emergency rescue services are some of the examples of ad hoc networks. Due to limitations of the transmission range, the nodes rely on each other for packet forwarding with the ultimate goal of delivering messages across the network. Thus, each node, apart from generating packets of its own, acts as a router and forwards packets on behalf of others. For routing purposes, the nodes resort to some sort of a routing algorithm which determines the path (route) along which a packet is to be forwarded from the source node to the destination node [2]. Most commonly used routing algorithms, though effective for non-real time packet delivery, fail to meet the delay bounds for packets generated from real-time applications. A good survey of routing protocols for ad hoc networks can be found in [3].

In this paper, we focus on real-time traffic that has critical delay requirements and demands that the data packets be delivered within certain delay bounds. It essentially demands certain quality of service assurance in terms of bandwidth, delay, and jitter. The limited radio bands (frequency spectrum) and interference due to route coupling make the task challenging. These challenges have led to investigations into various aspects that have enhanced network performance and efficiency. While quality assurance is necessary for real time traffic, it is also necessary to utilize the limited resources efficiently. Hence, it is important to strike a balance between the service quality and and the utilization of the resources.

Since timely delivery of packets is very crucial, average delay per packet is usually considered as the most important path selection criterion. The naive method would be to pick any path that offers the required bandwidth needed to support the traffic. However problem arises when new flows are admitted as they may adversely interfere with the ongoing sessions. At the same time it is not a good idea to block flows that could have been supported if paths were chosen wisely.

In this regard, we propose a routing protocol for carrying real time traffic that is based on the availability of resources (bandwidth) at every hop of a route. Alongside, resource reservation and residual bandwidth distribution schemes are proposed. The route selection algorithm coupled with the resource allocation mechanism can accommodate large number of concurrent real time flows with delay guarantees. The proposed scheme is distributed in nature and does not involve any centralized path computation which is usually very expensive. Moreover, the proposed scheme is a source-initiated on demand route acquisition system that does not require routing table exchanges or maintenance. In particular, the contributions are:

- We devise a route discovery scheme that is based on delay estimation along any route. The delay of a route is calculated by summing the delays at each hop. Of the multiple possible paths between the source and destination, the path that offers the minimum end-to-end delay without violating the the quality of on-going sessions is selected.
- We use a M/M/1 model to calculate the delay at each intermediate node. We consider the requirement of a new session in terms of the packet generation rate and evaluate if the residual bandwidth of the individual links can handle the demands of the new session.
- We propose a resource reservation mechanism that ensures that the quality of the on-going sessions do not

[1]This research was partially supported by ITT Advanced Engineering & Sciences.
Approved for Public Release; Distribution Unlimited: 88ABW-2010 3301 dated 16 June 2010.

degrade. This is achieved with the help of hard and soft reservation. The minimum amount of bandwidth needed to support the traffic is reserved while the residual bandwidth is distributed over the sessions to process packets faster.

- We conduct C based simulation experiments to validate the proposed schemes. We vary parameters like session generation rate, number of sessions requested, and network size. Performance of the proposed schemes is shown with respect to metrics like throughput, session blocking probability, average path length for a route, and average delay.

II. Proposed Routing Protocol

For any QoS aware routing, it is important that routes do not get disrupted frequently and the promised QoS is violated. Such guarantees can be made if we assume that the environment is less dynamic and topology remains the same during a session. An example could be a scenario where devices equipped with audio-video capturing capabilities are deployed at targeted locations. These nodes would monitor the area constantly and any event that triggers the sensors would lead to a request of path to send the information to a centrally placed node via the typical multihop communication.

A. Consideration for Algorithm Design

In this research, we restrict ourselves to limited movement of the nodes that form an ad hoc network. This network consisting of multiple nodes are confined to a two dimensional geographical area. We assume that nodes do not move to the extent that their neighbors change during packet transmission. We use the notion of delay bounds as a measure of quality of service. Average end-to-end delay is the metric for deciding the best path if multiple paths exist between a given source-destination pair. For every new session that a node generates, a new path discovery phase is initiated. Though multiple paths may exist between source and destination nodes, any intermediate node participating in route discovery is aware of just the minimum delay path through it.

B. Delay Calculation

We assume that the packets arrive at a node following a Poisson distribution and their processing times are exponentially distributed. Thus, the queue at any node can be modeled as a M/M/1 system [1]. The assumption of a M/M/1 model is justified as short messages have shown to follow a Markovian arrival process. Also, the service times of each packet is independent of each other. Moreover, it is easy to analyze M/M/1 systems because of their mathematical tractability.

To calculate delay, we assume that each node generates request for new sessions which have their traffic generation rates and packet sizes. If packet arrival rate is λ and service rate is μ, then delay can be calculated using Little's Law. The average delay is estimated as $\frac{1}{\mu-\lambda}$. This delay is inclusive of service time and queuing delay. The buffered packets are served as per FIFO (first in first out) policy at a rate equivalent to the channel capacity of the outgoing link.

The delay incurred by any packet at node i is given by

$$\mathcal{D}_i = \frac{1}{\mu - \lambda}.$$

The total delay along path from a source to a destination can be obtained by adding up delays incurred at each hop along the path i.e.,

$$\mathcal{D}_{total} = \sum_{i \in P} \mathcal{D}_i$$

where P is the set of nodes comprising the path from source to destination.

C. Resource Reservation Techniques

The delay calculation takes into account the service rate which is essentially an estimation of available bandwidth. The available bandwidth needs to be more than the data rate being injected into the system by the source. This assurance can be achieved by reserving bandwidth. There are typically two types of reservation mechanisms.

1) Soft Reservation: In this case, a reservation is made for a given period of time. As soon as resources are allocated to a flow, an associated timer is started at the end of which the resources are released. However if updates are sent before expiration then the timer is set accordingly and the reservation period may be extended.
2) Hard Reservation: Though the reservations are made in a similar way, hard reservation needs explicit release message in order to release the reserved resources. There is no lifetime or timer associated with hard reservation.

The proposed schemes use a combination of both soft and hard reservation. The minimum bandwidth required to support the session is allocated permanently until the session is over while the residual bandwidth allocation varies as sessions begin or end between a given pair of nodes.

D. Algorithm Description

Here we describe the algorithm for selecting minimum delay path on a per hop basis. The delay is estimated at each node along the path with sufficient resources and the path with minimum end-to-end delay is selected. The formal algorithm is presented in Fig. 1 with the detailed steps discussed next.

1) A source node creates a request by specifying the traffic descriptors like the packet arrival rate, λ packets/second and the packet size, c bits/packet.
2) The source node then checks its outgoing links for available bandwidth. All neighbors receive the request, but only the ones with sufficient resources along the link will further send the request to their neighbors.
3) While doing so, the source node calculates the delay along each outgoing link and then forwards the request to its neighbors. In this way the delay incurred upto a node along the path is known. In case the link capacity is not enough to support the session, the delay is set to infinity and such requests are not sent any further.

Min. Delay Route Algorithm

For every node N_i, setup neighbor list $L(N_i)$;
For every neighbor N_j of node N_i, set channel capacity C_{ij};
A request for transmission with packet arrival rate= λ_k is initiated by node N_k;
for all outgoing links OL_k through N_k **do**
  if $bandwidth_{available}(OL_k) > \lambda_k$ **then**
    Calculate $delay = \frac{1}{capacity_{available}(OL_k) - \lambda_k}$;
    Add its own id in the request packet;
    Cache the request packet;
  else if $bandwidth_{available}(OL_k) < \lambda_k$ **then**
    $delay = \infty$
  end if
  Forward the request packet to the neighbor;
end for
Check the cache of every node N_r that receives the request;
if This request is received for the first time **then**
  for all outgoing links OL_r through N_r **do**
    if $bandwidth_{available}(OL_r) > \lambda_k$ **then**
      Calculate $delay = delay + \frac{1}{bandwidth_{available}(OL_r) - \lambda_k}$;
      Add its own id in the request packet;
      Cache the request packet;
    else if $bandwidth_{available}(OL_k) < \lambda_k$ **then**
      $delay = \infty$
    end if
    Forward the request packet to the neighbor;
  end for
else if This request is present in the cache **then**
  if the delay in the new request is less than the delay in cache **then**
    for all outgoing links OL_r through N_r **do**
      if $bandwidth_{available}(OL_r) > \lambda_k$ **then**
        Calculate $delay = delay + \frac{1}{bandwidth_{available}(OL_r) - \lambda_k}$;
        Add its own id in the request packet;
        Replace the cache request packet;
      else if $bandwidth_{available}(OL_k) < \lambda_k$ **then**
        $delay = \infty$
      end if
      Forward the request packet to the neighbor;
    end for
  end if
end if

Fig. 1. Minimum Delay Route Algorithm

4) A node that receives the request for the first time does the same thing as the source node. It also calculates the delay along each outgoing link with sufficient resources, and adds this delay to the delay incurred in reaching this node from the source and forwards the request further.
5) If the node has already forwarded a request earlier, it checks to see if the new path along which it receives the request has a lower delay to offer. If so, then it repeats the above step and updates its cache with the new request packet.
6) If the receiving node is the destination node then it does not forward the request further.

In the proposed algorithm, the service for the channel between two adjoining nodes needs to be estimated in order to calculate the delay. This is the exponential service rate at which packets from the sending node's queue will be processed if the session is admitted. The next section describes the steps involved for service rate estimation.

E. Resource Allocation

Though packet arrival process is bursty in nature, we can always find the long term average arrival rate. As long as the bandwidth is enough to support the long term average, we assume that the quality of service is acceptable. Hence, any new flow being requested to be admitted into the system should be ensured a minimum guarantee on the bandwidth. Hence bandwidth equivalent to the packet arrival rate as specified in the request can be allocated for the session till it ends. This hard reservation of resources are dedicated to a flow until the flow ends.

Larger bandwidth results in faster processing of the packets and hence delay is less. Thus a higher service rate is always preferred. At any instance, there could be some channel bandwidth that has not been assigned to any ongoing session; this is known as the residual bandwidth. We distribute this residual bandwidth proportionally amongst all ongoing sessions. This assignment is like soft reservation. However, allocation of a part of the residual bandwidth is not for any pre-determined time period. Hence, the extra allocated bandwidth can be withdrawn in order to accommodate any new flow that would share the residual bandwidth.

To summarize, all flows are admitted if and only if the delay requirements can be met. This is achieved through hard reservation. If residual bandwidth is available, then it is shared in a proportional manner. The algorithm for resource allocation is explained in Fig. 2.

III. Simulation Model and Results

To validate the proposed protocol, we conducted extensive simulation experiments. We considered a set of nodes that were randomly distributed over a square region. For simplicity, the nodes were assumed to be identical with respect to their transmission and receiving power. The nodes remain in their positions so that the path is not disturbed during a session. The channel capacity between any pair of nodes is assumed to be constant. We consider that the nodes generate flows that are real time data packets only. The packet arrival process is assumed to be Poisson and the service time is exponentially distributed. Packets are lost due to channel errors. Since we do not deal with the medium access control (MAC) layer, we assume a MAC that accounts for collision free environment.

Resource Allocation Algorithm

Calculate channel capacity C_{AB} between nodes N_A and N_B;
Set residual bandwidth $RB_{AB} = C_{AB}$;
if A new session S_i is allowed through nodes $N_A \rightarrow N_B$ with packet arrival rate λ_i **then**
    $RB_{AB} = RB_{AB} - \lambda_i$;
    $\mu_i = \lambda_i + \frac{\lambda_i}{\sum_{k=1}^{n} \lambda_k} * RB_{AB}$; where n is total number of sessions through this pair of nodes
    for all ongoing sessions through this pair of nodes **do**
        $\mu_l = \lambda_l + \frac{\lambda_l}{\sum_{k=1}^{n} \lambda_k} * RB_{AB}$;
    end for
end if
if An ongoing session S_j terminated between nodes $N_A \rightarrow N_B$ with packet arrival rate λ_j **then**
    $RB_{AB} = RB_{AB} + \lambda_j$;
    for all ongoing sessions through this pair of nodes **do**
        $\mu_l = \lambda_l + \frac{\lambda_l}{\sum_{k=1}^{n} \lambda_k} * RB_{AB}$;
    end for
end if

Fig. 2. Resource Allocation Scheme

TABLE I
SIMULATION PARAMETERS

Number of nodes	60,80,100
Total area	200x200
Transmission range	40
Audio Packet generation rate	20 packets/sec
Audio Data packet size	100 bytes
Video Packet generation rate	[25-500] packets/sec
Video Data packet size	500 bytes

A. Simulation Parameters

The simulation is carried for three different network sizes with 60, 80 and 100 nodes. The nodes are randomly placed over a 200 $\times$ 200 area and the packets they generate are destined for a centrally located sink. The transmission range is 40 for each node. The rate at which requests for packet transmission are generated is varied from as low as 0.002 requests/second to 1 request/second. The request is made for either audio or video traffic. Data for an audio session has packet generation rate of 20 packets/second with each data packet of size 100 bytes. Video traffic has packet generation rate between 25 and 500 packets/second with packet size of 500 bytes. All packets are generated as per the Poisson model and packet service times are exponentially distributed. Hence, all nodes are modeled as M/M/1 systems.

Table I summarizes the simulation parameters used in the experimental setup.

B. Metrics of Interest

The simulation is carried to study the behavior of the proposed routing scheme with respect to various metrics. Throughput, session blocking probability, average number of hops, fraction of paths through minimum hop, and average delay incurred are some of the metrics that are used to evaluate the routing protocol. Let us formally define these metrics.

- Throughput: Throughput is defined as the average number of packets received successfully at the destination per unit time. This is obtained by counting the total number of packets successfully reaching the receiver and dividing by the simulation time.
- Session Blocking Probability: It is the ratio of denied session requests to total number of requests made. The sessions are denied due to insufficient resources in the network.
- Packet Dropping Probability: Packets in a wireless network can be dropped either due to lossy channel or due to buffer overflow. Here we measure the packet dropping probability as the ratio of packets dropped due to buffer overflow to total number of packets generated.
- Average Path Count: It is the average number of hops taken from source to destination. All requests are routed through the minimum end-to-end delay path. This path is decided based on bandwidth availability and may not be the minimum hop path. This way, we can find the average number of hops taken to reach the destination.
- Fraction of Paths Through Minimum Hop: As mentioned earlier for the average path count, the packets may be routed through a path that has more hops. This metric helps in estimating what percentage of total paths take the minimum hop path.
- Average Delay: This is the average delay incurred by a packet when it travels from a source to the destination. This is obtained by dividing the total delay by total number of successful packets.

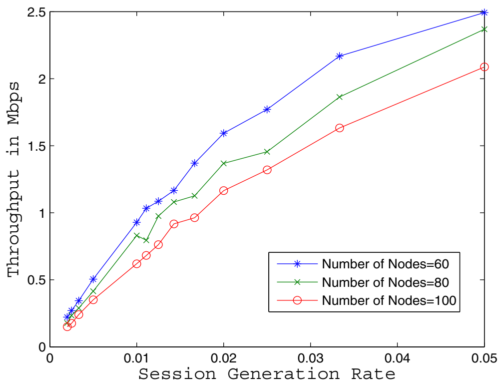

Fig. 3. Throughput with increasing session generation rate

C. Throughput

Throughput of a network is a measure of its packet delivery efficiency. The plot presented in Fig. 3 shows the throughput for varying session generation rates. When requests for session are made at a higher rate, i.e., the time between two successive requests for route discovery is less, the throughput is higher. We vary the rate from 0.002 sessions/second to 0.05 sessions

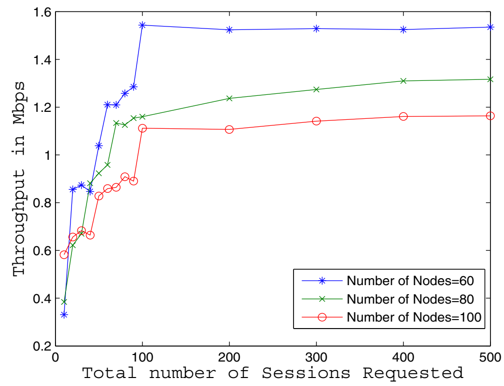

Fig. 4. Throughput for a fixed session generation rate of 0.02 sessions/sec.

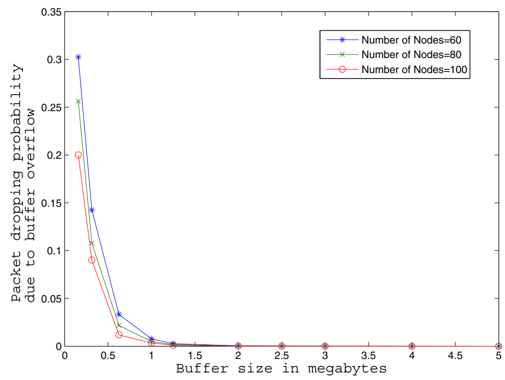

Fig. 6. Packet dropping probability due to buffer overflow

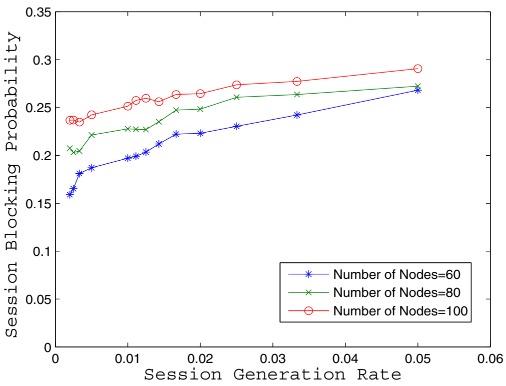

Fig. 5. Session blocking probability with session generation rate

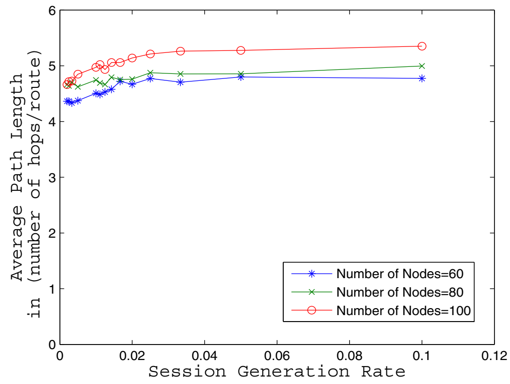

Fig. 7. Average Number of hops with increasing session generation rate

per second. 500 requests were made one after the other for a given session generation rate and the corresponding throughput is obtained for that particular session generation rate. The throughput increases from approximately 0.2 Mbps to about 2.5 Mbps. From the plots, we can infer that for higher request generation rate the packet delivery rate scales well. Hence the system is expected to have good performance with respect to request rate.

Fig. 4 shows the effect of increasing the number of requests at a given session generation rate. The rate of requests is fixed at 0.02 sessions/second and number of requests made is varied form 10 request to 500 requests. The throughput increases initially till 100 sessions have been requested. At around 100 requests the system reaches its saturation.

In the same plot, the simulation result is observed for different network sizes. The throughput is slightly higher for smaller network size. A smaller network has higher saturation level. It is due to the smaller routes taken in a smaller network. Thus we see that a systematic route selection results in higher throughput for the network.

D. Session Blocking Probability

We study the blocking probability of the proposed scheme. As more and more requests are generated in a given time, the resource limitations would deny admission to some of the requests. The blocking probability increases for higher request rate as seen in Fig. 5. However even for high rates equivalent to one request every 20 seconds i.e. at a rate of 0.05 sessions/second the blocking probability is less than 30 percent for the different network sizes. For most of the applications the request is not so frequent and the blocking probability varies between 0.15 and 0.25 for different network sizes. Hence more and more requests are denied and the network utilization suffers if the routes are not chosen strategically.

E. Packet Dropping Probability

Due to finite buffer size, packets will be lost due to buffer overflow. We study the packet drop probability for different buffer sizes. We vary the buffer size from 0.15625 megabytes to 5 megabytes. Fig. 6 shows that the packet drop probability for a buffer size of 0.625 megabytes is 4% and drops to nearly 0% for a buffer size of 1 megabyte or higher.

F. Average Path Count

Fig. 7 shows the average number of hops taken per request. This value does not change with the request rate. The plots show that for a network size of 60 nodes the path length varies between 4.4 hops/route to 4.7 hops per/route. Similarly, for 80

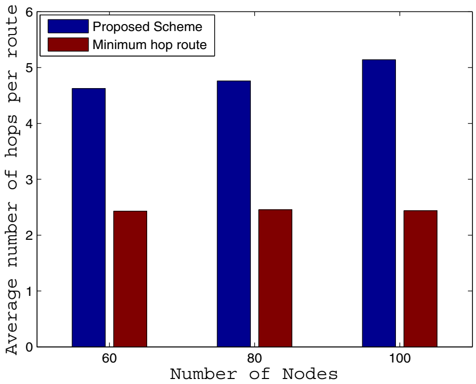

Fig. 8. Average Number of hops for the proposed and minimum hop scheme

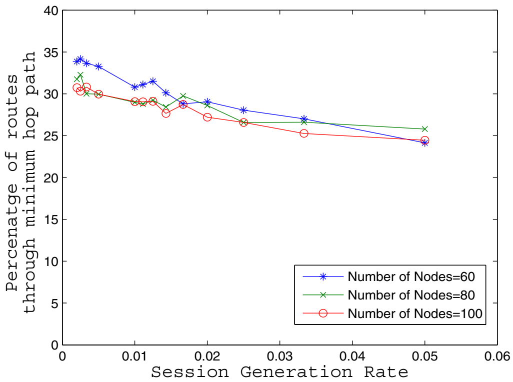

Fig. 9. Percentage of paths through minimum hop

and 100 nodes the range is between 4.6 to 4.9 and between 4.6 to 5.2 hops/route respectively.

The number of hops taken along a path is more for higher request rate. The shorter paths with sufficient resources to support the requests are exhausted soon and for further requests it becomes necessary to take longer paths to ensure quality.

Fig. 8 shows that the tradeoff due to our proposed scheme. We observed that the path length of a route is higher as compared to the minimum hop path scheme. Our routing protocol selects routes that offer the least delay irrespective of the number of hops. This also helps in distributing the load over the network.

G. *Percentage of paths through minimum hop*

As seen earlier in Fig. 8 the path count is higher for our scheme. The result in Fig. 9 shows that about one-third of the paths happens to be through the minimum hop path irrespective of the network size. This value further drops for higher request rates.

H. *Average Packet Delay*

Since we consider delay guarantees for assuring quality of service, it is very essential to see delay performance of the scheme. Fig. 10 shows the delay performance for our

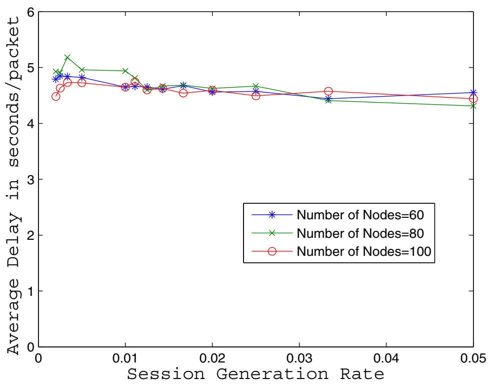

Fig. 10. Average packet delay with increasing Session Generation Rate

scheme. There is a small variation in the delay for different network size for lower session generation rate. However with increasing session generation rate the delay performance is same irrespective of the network size as well.

Thus, we see that varying parameters like session generation rate and network size do not affect the delay performance adversely. The delay performance is consistent and hence we successfully achieve delay guarantee.

IV. Conclusions

In this paper, we propose a delay based routing protocol for ad hoc networks that support real time applications. The proposed routing protocol estimates the delay of a new session based on resource availability. If resources are available, i.e., there is enough bandwidth, then a new session is admitted and a route that yields the minimum delay from the source to the destination is found. Delay at each hop is determined using the M/M/1 model; the end-to-end delay is calculated by summing the delays at each hop. For better delay performance, the routing protocol is complimented with adaptive resource management. Resources are reserved for sessions that have already been admitted and the residual bandwidth is distributed proportionally to aid faster processing of packets. In this way, the best possible quality of service is made available to the data traffic at any time. All session requests were dealt individually as per the traffic descriptors like packet generation rate and packet size. Simulation results reveal that the proposed routing scheme adheres to the delay bounds. The results also show that the delay variation is very small irrespective of varying parameters like number of nodes and traffic generation rate. Hence, the target delay is achieved using our approach.

References

[1] D. Bertsekas and R. Gallager, “Data networks (2nd ed.)”, ISBN 0-13-200916-1, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[2] D. Johnson and D. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, Mobile Computing, Vol. 353, 1996.

[3] C. Lei and W.B. Heinzelman, “A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks”, IEEE Network Magazine, Volume 21, Issue 6, 2007, pp. 30-38.

[4] R. Ramanathan, and J. Redi, “A brief overview of ad hoc networks: challenges and directions”, IEEE Communications Magazine, May 2002, Vol. 40, Issue 5, pp. 20–22.